

14.13 Testkonzept

Das Testkonzept definiert wie, was, wann und wieviel getestet wird.

14.13.1 Rahmenbedingungen

Beispiel:

Das zu testende System ist der yx im Zusammenspiel mit den Sensoren. Dabei wird hauptsächlich nur das neue Feature "Firmware Manager", welches ich während der IPA umsetzte, getestet. Zudem werden auch noch Teile der Device Control getestet (altes Upgrade von Sensoren). Die yx Sensoren sind auch Teil des zu testenden Systems, da getestet wird, ob diese aktualisiert werden können.

Im Rahmen der IPA beschränkt sich das Testing auf Black-Box-Tests und JUnit Tests. Getestet wird ausschliesslich in der neusten Version von Google Chrome.

Abnahme Tests werden während der IPA keine durchgeführt, da diese bei uns in der Firma erst beim Release gemacht werden und das Feature, welches ich implementiere, nicht direkt released wird. Auch werden keine Frontend-Tests geschrieben, da diese bei uns in der Firma eher weniger Priorität haben und es den Rahmen der IPA übersteigen würde.

14.13.2 Testumgebung

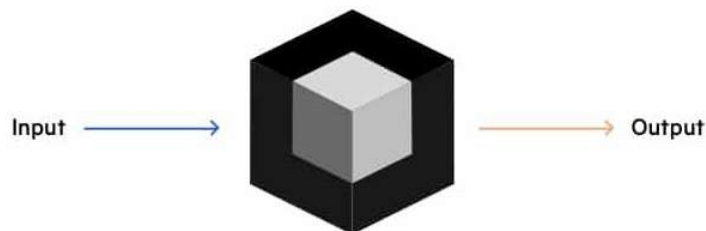
Beispiel:

Zum Testen wird der aktuelle Stand des xy des Firmware Managers (epic/S2- 4040-firmware-manager) verwendet. Dieser wird von den Testern (Name Vorname) lokal in der Entwicklungsumgebung ausgecheckt. Lokal wird dann der aktuelle Stand aufgesetzt (siehe Anleitung: xy Development Setup [7]) und konfiguriert. In der yx Runtimekonfiguration muss das Property "max_body_size_mb" mit dem Wert 60 unterhalb vom Property "show_screen_shot" hinzugefügt werden. Gestartet werden muss danach die Datenbank, der Dev Keycloak, das yx Backend und yx Web.

Die Firmware-Pakete, welche für die manuellen Tests gebraucht werden, sind in einem Ordner auf dem yx Share, welcher mit der Testperson definiert wurde. Die Firmware Pakete sind wie in den Tests beschriftet.

14.13.3 Testmethoden

Als Testmethode wird die Black Box Methode verwendet. Diese bietet sich sehr gut an, da hier die Tests ohne Kenntnisse der inneren Funktionsweise des zu Testenden Systems durchgeführt werden. So werden nur die nach aussen sichtbaren Verhalten der Systeme getestet. Die Tests werden von Testern durchgeführt und anschliessend bewertet. Die Testpersonen sind im Kapitel XY Projektorganisation ersichtlich. Falls Mängel während dem Test zum Vorschein kommen, wird entschieden ob Massnahmen nötig sind und welche Verbesserungen vorgenommen werden.



Black Box

Abbildung SEQ Abbildung * ARABIC 5: Black Box Testmethode

14.13.3.1 Black-Box-Tests

Beispiel:

Die Black-Box-Tests werden anhand der Use Cases entwickelt und müssen alle Szenarien abdecken. Black-Box-Tests sind auf einer hohen Ebene der Applikation angesiedelt, da sie nur das nach Aussen sichtbare Verhalten testen. Die Black-Box-Tests können in diesem Fall als manuelle Systemtests angesehen werden. Es werden die Funktionalitäten der Applikation aus Benutzersicht getestet, ohne auf die interne Struktur der Applikation einzugehen.

14.13.3.2 JUnit Tests

Beispiel:

JUnit Tests werden automatisiert durchgeführt. Diese Art von Tests ist viel tiefer als die Black-Box-Tests angesiedelt und testet die einzelnen Funktionen von Codekomponenten. Es wird getestet, ob die Logik eines einzelnen Teils des Programms korrekt funktioniert und das Resultat im erwarteten Zustand ist oder ob etwas in der richtigen Reihenfolge ausgeführt wurde. In meinem Fall will ich für den Backendcode eine Testabdeckung von 70% erzielen und die DB Tests sollten sowohl für MSSQL als auch für PostgreSQL funktionieren.

14.13.4 Testmittel

Mittel zur Ausführung der Black-Box-Tests:

- Tester (Name Vorname)
- Entwicklungs-PC
- Laufende xy Instanz (DB, Frontend, Backend, Auth)
- Google Chrome (89.0.4389.58)
- Testfälle und Testprotokoll

Mittel zur Ausführung der JUnit Tests:

- Entwicklungs-PC (siehe Spezifikation im Kapitel 4.2.1)
- IBEX Source Code (JUnit Tests)
- Laufende Datenbank
- Auszufüllendes Testprotokoll

14.13.5 Was wird nicht getestet

Die manuellen Systemtests testen nicht das korrekte Auslesen der Firmwaredaten in jedem einzelnen Spezialfall (Version, Pre-Release or Latest, Post 3.7 oder Pre 3.7, die verschiedenen Generationen), da all diese Spezialfälle in den JUnit Tests abgedeckt werden sollten und deshalb bereits getestet sind.

14.13.6 Testziele

Beispiel:

Durch die verschiedenen Arten von Tests soll die Qualität sichergestellt werden. Das Testing erlaubt mir, möglichst früh Fehler zu erkennen und diese noch beheben zu können. Die JUnit Tests sollen die grundlegende Logik und Funktionalität validieren. Durch die Black-Box-Tests kann anschliessend der wirkliche Einsatz (nach Use Cases) getestet und verstanden werden, ob die Funktionen benutzerfreundlich und intuitiv sind.

14.13.7 Testorganisation

Beispiel:

Das Testing wird im Rahmen meiner IPA von zwei Personen durchgeführt. Die JUnit Tests werde **ich** selbst schreiben, ausführen und protokollieren. Die Black-Box-Tests werden jedoch von einer

Person durchgeführt, die den geschriebenen Code nicht kennt. Dies ist in diesem Fall Name Vorname.

14.13.8 Kriterien für Erfolg / Misserfolg

Beispiel:

Ein Test gilt als erfolgreich, wenn alle erwarteten Resultate, welche in einem Testfall definiert wurden, erfüllt sind. Werden die Resultate nicht erfüllt, so zählt der Test als fehlgeschlagen.

14.13.9 Fehlerbehandlung

Kann ein Test nicht durchgeführt werden oder das erwartete Resultat wird nicht erfüllt, so muss ein Fehlerprotokoll ausgefüllt werden. Das Fehlerprotokoll befindet sich immer am Ende eines Testfalls und muss nur ausgefüllt werden, wenn ein Fehler aufgetreten ist. Im Fehlerprotokoll wird das aufgetretene Problem und mögliche Ursachen beschrieben. Das Problem wird dann nach dem Testen behoben und der Test nochmals durchgeführt. Ist der Test nun erfolgreich, gibt es kein Problem mehr. Kann ein Test jedoch nicht mehr im Rahmen der IPA zum Laufen gebracht werden, so wird dies dokumentiert, begründet und dann nach der IPA behoben.

14.13.10 Testrahmen

Bevor angefangen wird zu testen, müssen bestimmte Voraussetzungen erfüllt sein. Als erstes muss das Testkonzept ausgearbeitet sein. Die Migration muss gestartet oder schon abgeschlossen sein, damit die Tests durchgeführt werden können. Die einzelnen Tests haben jeweils spezifischere Anforderungen, diese werden im Kapitel *yx Testfälle* näher beschrieben.

Die Tester bekommen jeweils ein Testprotokoll, welches sie ausfüllen müssen, wobei sie die Tests in vier Klassen einteilen. Die Klassen sind wie folgt:

ID	Mangel ausmass	Beschreibung
TC-1	Kein Mangel	Hat keine Mängel vorzuweisen.
TC-2	Unwesentlicher Mangel	Der Mangel beeinträchtigt die Funktionalität nicht, wird dennoch mit dem Administrator besprochen.
TC-3	Kleiner Mangel	Der Mangel beeinträchtigt die Brauchbarkeit und kann störend sein, jedoch funktioniert das System.
TC-4	Grosser Mangel	Das System ist funktionsfähig, aber nicht in vollem Ausmass und mit vielen Fehlern.

Ist ein Test mit unwesentlicher oder kleiner Mangel klassifiziert worden, wird der Fehler sofort behoben und es wird erneut getestet.

Falls ein Test mit grosser Mangel klassifiziert wird, muss mit dem Auftraggeber abgeklärt werden wie das weitere Vorgehen ist. Der Fehler wird nach Absprache behoben und es wird erneut getestet.

14.13.11 Testobjekte

Nachfolgend sind die Testobjekte aufgeführt, auf welche zugegriffen werden muss, um die Tests durchzuführen.

ID	Objekt	Beschreibung
TO-1	fwairz0031	1. Firewall Server
TO-2	fwairz0032	2. Firewall Server
TO-3	fwairz0031	Firewall Cluster
TO-4	fwairz0031_BackupLAN	Firewall Ruleset
TO-5	Logs	Firewall Logs
TO-6	Kundennetze	Kundennetze für die Verbindung zu testen
TO-7	rtaizk0007v11212	Switch in dem Backup Netzwerk
TO-8	wvairz0249	Windows Jumpserver
TO-9	abairz0015	Multi Domain Server

14.13.12 Testfälle

Text

Vorbereitungsschritte**Einloggen im System XY:**

1. Google Chrome Browser öffnen und die Adresse "localhost:4200" eingeben
 - a. Zum Einloggen mit dem Admin-Benutzer folgende Daten eingeben:
 - Username: admin
 - Passwort: pass
 - b. Zum Einloggen mit dem Benutzer, welcher nicht alle Rechte hat:
 - Username: test
 - Passwort: test

Testbenutzer, der nicht alle Rechte hat, erstellen:

1. Einloggen mit Admin Benutzer
2. Im Hauptmenu auf "Configuration" wechseln
3. Von den verschiedenen Kacheln "User Management" auswählen
4. Testbenutzer mit dem Namen test und dem Passwort test erstellen
5. Dem Benutzer über eine Gruppe nur das Recht "Configure Sensors" geben (siehe Anleitung: xy+User+and+group+management.doc [9])

Navigieren zum Firmware Manager:

1. Einloggen
2. Im Hauptmenu auf "Configuration" wechseln
3. Von den verschiedenen Kacheln "Device Control" auswählen
4. In den Tab "Firmware Manager" wechseln

Testfälle Nr.:	TF-1	
Abgedeckter Use Case:	UC-5	
Testumgebung:	yx Chrome (89.0.4389.58)	
Funktionalität:	Firmware hinzufügen	
Vorbereitungsschritte:	1. Ein gültiger Benutzer ist am System eingeloggt 2. Der Benutzer befindet sich im Firmware Manager	
Testschritte		
Nr.	Aktion	Erwartetes Ergebnis
01	Klicken Sie den Button "Add"	Ein Dialog, um eine Firmware hinzuzufügen, öffnet sich
02	Wählen Sie die Firmware "upgrade_3.9.10_2ndGen_post3.7.xfw" von dem Computer aus	Der Name (upgrade_3.9.10_2ndGen_post3.7.xfw) des Pakets wird im Upload Dialog angezeigt
03	Klicken Sie auf den Button "Submit"	Der Dialog schliesst sich, nachdem die Firmware gespeichert wurde und eine Erfolgsmeldung erscheint.
04		
05		
06		
07		
08		
09		
10		

14.14 Phasenfreigabe Realisierung

Mit der Unterschrift des Fachvorgesetzten wird bestätigt, dass die Konzeptphase abgeschlossen ist und mit der Realisierung begonnen werden kann.

Datum	Name / Rolle	Unterschrift
xx.xx.2022	Name Vorname Lernender	
xx.xx.2022	Name Vorname Vorgesetzte Fachkraft	